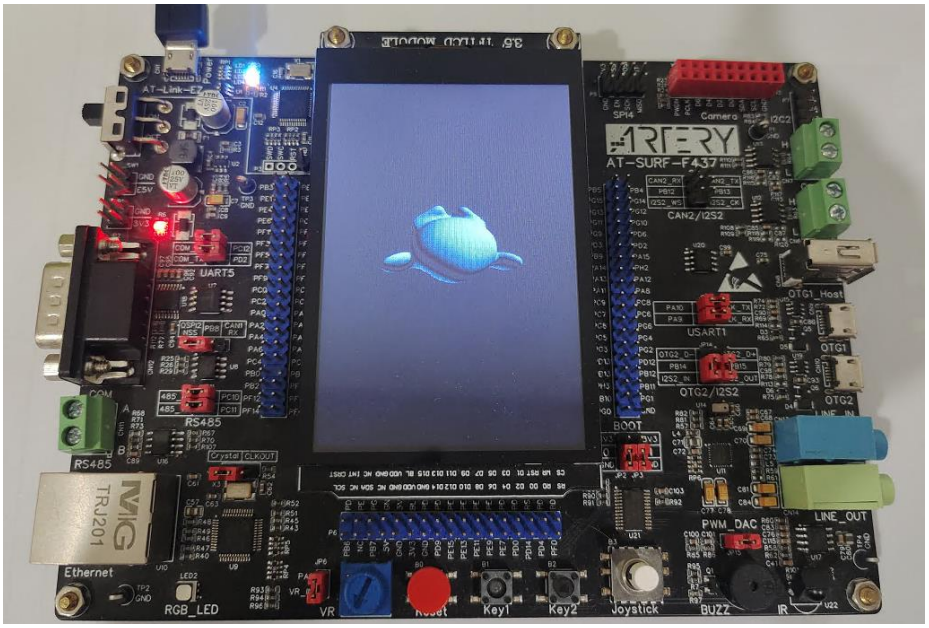
SC0108
Sample Code

EmberGL On AT32F437

Introduction

This sample code demonstrates ow to use EmberGL on AT32F437 series.

Video demo link:
https://b23.tv/ieMLePz



Applicable products:

Product series	AT32F435xx
	AT32F437xx

A list of major peripherals used:

Peripherals	XMC
	DMA
	USART

1 EmberGL overview

EmberGL (Ember Graphics Library) is a low-level open source graphics library, similar to OpenGL/DirectX/Vulkan, designed for real-time 2D/3D rendering on MCUs and other memory constrained non-GPU systems. The graphic API has been specifically designed for such systems, utilizing modern techniques to be able to maximize rendering performance within tight memory budgets, while providing a lot of flexibility and customizability. The library can be also useful on other targets with more generous budget, for projects which benefit from software rasterization, and can be compiled for example with Visual Studio and GCC. Because EmberGL is a low-level library, it provides only the core rendering functionality of flexible and efficient triangle rasterization, along with supporting components and a set of display drivers. These low-level features can be used either for direct application development or development of efficient higher level graphics libraries, such as GUI libraries or 3D engines.

The library features a tile-based software rasterizer, which enables flicker-free rending without requiring RAM for the entire display frame and depth buffers, thus expands the applicability of the library to a wider set of devices and projects. Tile-Base Rendering (TBR) architecture are commonly used on mobile devices and also on some desktop GPUs mainly due to the memory bandwidth benefits (more info on ARM Developer website). The TBR engine of EmberGL can be customized for wide range of memory and performance requirements by configuring the rasterizer properties such as the tile size, depth buffer format, intermediate tile pixel format, etc.

EmberGL supports various fixed-function pipeline features, such as a set of depth tests, triangle culling modes, triangle interpolation modes, etc. In addition to the fixed-function features, the library also supports C++ programmable blending and vertex & pixel shading stages for custom geometry and lighting effects. To obtain high performance while supporting a flexible set of features, the rasterizer extensively utilizes C++ templates to generate optimized rasterizers for each set of utilized features during program compilation time. This eliminates any unused feature branching and tightly embeds shader code to the rasterizer, resulting in optimized rasterizers for each used combination of features and shaders.

In addition to the rasterizer, EmberGL provides a growing set of optimized display drivers to effectively deliver the rasterized pixels to the display. For example, the library contains optimized display driver for popular ILI9341 display with DMA support. The set of drivers can be extended by implementing a narrow device interface, which hooks new drivers to the rasterizer, where the main implementation focus is the device initialization and pixel data transfer without having to worry about the rasterizer complexities.

EmberGL source code address:
https://github.com/EmberGL-org/EmberGL

2 Quick start

2.1 Hardware resources

1) AT-SURF-F437 evaluation board

2.2 Software resources

- SC0108_SourceCode

2.3 Example case

Using software:

Open corresponding project under “**utilities\EmberGL_Demo**”, compile and download it to the evaluation board.

- A rotating 3D graph is displayed on LCD

3 Revision history

Table 1. Document revision history

Date	Revision	Changes
2023.02.16	2.0.0	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein.

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license grant by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement of any patent, copyright or other intellectual property right.

Purchasers hereby agrees that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any air craft application; (C) any automotive application or environment; (D) any space application or environment, and/or (E) any weapon application. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and is solely responsible for meeting all legal and regulatory requirement in such use.

Resale of ARTERY products with provisions different from the statements and/or technical features stated in this document shall immediately void any warranty grant by ARTERY for ARTERY products or services described herein and shall not create or expand in any manner whatsoever, any liability of ARTERY.